Conclusions

A Newton-Krylov method has been applied to both two-dimensional and three-dimensional unstructured Euler codes. Results are presented for various geometries, and comparisons of solution times have also been shown for the various methods examined. A method used to select an appropriate perturbation for the finite-difference of residuals has also been presented which yields consistent convergence levels, regardless of the mesh size.

The Newton-Krylov method converged faster than the point Gauss-Seidel method in subsonic cases, but suffered for transonic conditions due to CFL number constraints for the early iterations until the shock position stabilized. It was found that this problem could be overcome by using the Gauss-Seidel scheme in early iterations and switching to the Newton-Krylov scheme only after the residual had dropped by a specified amount. A mesh sequencing approach using the point Gauss-Seidel method on the coarse grid levels and the Newton-Krylov method on the finest grid level has also been shown to be effective. For both of these schemes, the rapid convergence of the Newton-Krylov method is obtained in the final stages of the iterative process without incurring the penalties associated with using this methodology in the initial iterations.

The approximate method for forming the matrix-vector product was also compared with an exact method. The two compared favorably, with a slight reduction in computer time for the exact method, although this was at the expense of extra memory requirements.

Acknowledgments

The authors would like to thank Henry Jones, for providing the geometry for the Apache helicopter and Javier Garriz, who provided meshes for the 3-D computations.

References

[1]Anderson, W. K., and Bonhaus, D., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids." *Computers and Fluids*, Vol. 23, No. 1, pp. 1-21, 1994.

[2]Anderson, W. K., "Grid Generation and Flow Solution Method for the Euler Equations on Unstructured Grids," *J. Comp. Phys*., Vol. 110, No. 1. January 1994.

[3]Anderson, W. K., Rausch, R. D., and Bonhaus, D., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids", AIAA 95-1740, June 1995.

[4]Barth, T. J., and Linton, S. W., "An Unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation." AIAA Paper 95-0221 (1995).

[5]Bonhaus, D., "An Upwind Multigrid Method for Solving Viscous Flows on Unstructured Triangular Meshes," M.S. Thesis, George Washington University, Aug. 13, 1993.

[6]Brown, P. N., and Saad, Y., "Hybrid Krylov Methods for Nonlinear Systems of Equations," *SIAM J. Sci. Stat. Comput*., Vol. 11, pp. 450-481, (1990).

[7]Cai, X.-C., Gropp, W. D., Keyes, D. E., and Tidriri, M. D., "Newton-Krylov-Schwarz Methods In CFD", in "Proceedings of the International Workshop on Numerical Methods for the Navier-Stokes Equations" (Heidelberg, October 1993), Vieweg Verlag, Braunschweig, 17 (1994).

[8]Cuthill, E., and McKee, J., "Reducing The Band Width Of Sparse Symmetric Matrices." Proc. ACM National Conference, 157 (1969).

[9]George, A., and Liu, J. W., Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall Series in Computational Mathematics, Englewood Cliffs, N. J., 1981.

[10]Hackbusch, W., Iterative Solution of Large Sparse Systems of Equations, Springer-Verlag, New York, 1994.

[11]Johan, Z. and Hughes, J. R., "A Globally Convergent Matrix-Free Algorithm for Implicit Time-Marching Schemes Arising in Finite Element Analysis in Fluids," *Computer Methods in Applied Mechanics and Engineering*, Vol. 87, pp. 281-304, 1991.

[12]Knoll, D. A., and McHugh, P. R., "Inexact Newton's Method Solutions To The Incompressible Navier-Stokes And Energy Equations Using Standard And Matrix-Free Implementations." AIAA Paper 93-3332 (1993).

[13]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes." *J. Comp. Phys*., Vol. 43, 1981.

[14]Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm For Solving Nonsymmetric Linear Systems." *SIAM J. Sci. Stat. Comput.*, Vol. 7, pp. 856 (1986).

[15]Saad, Y., "Krylov Subspace Methods on Supercomputers," *SIAM J. Sci. Stat. Comput*., Vol. 10, pp. 1200-1232 (1989).

[16] Schmitt, V., and Charpin, F., "Pressure Distributions On The ONERA M6 Wing At Transonic Mach Numbers." *Experimental Data Base for Computer Program Assessment*, AGARD-AR-138, B1-1 (1979).

[17]Van Leer, B., "Flux-Vector Splitting For The Euler Equations." ICASE Technical Report 82-30 (1982); also: Lecture Notes in Physics, 170, 501 (1982).

[18]Venkatakrishnan, V., and Mavriplis, D. J., "Implicit Solvers for Unstructured Meshes," *J. Comp. Phys*., Vol. 105, No. 1, pp. 83-91, June, 1993.

[19]Wigton, L. B., Young, D. P., and Yu, N. J., "GMRES Acceleration Of Computational Fluid Dynamics Codes." AIAA Paper 85-1494 (1985).

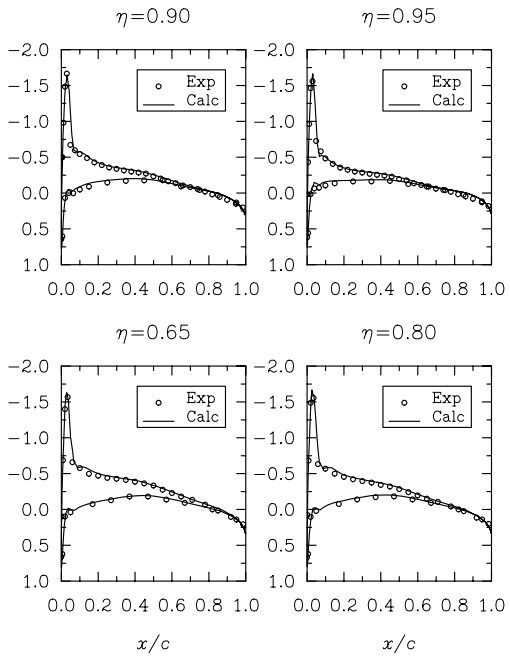

Fig. 10. Pressure distributions for ONERA M6 wing at several span stations. $M_\infty = 0.699$, $\alpha = 3.06^\circ$.

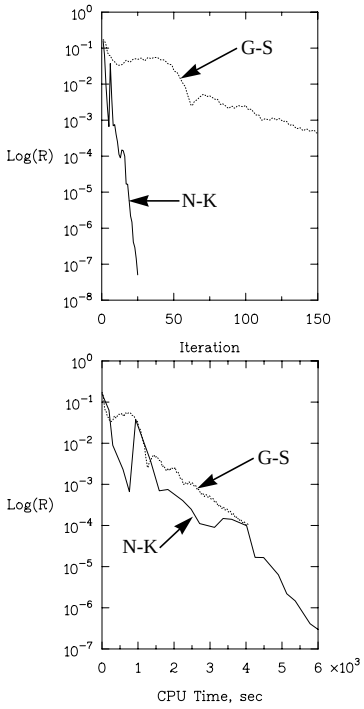

Fig. 12. Convergence histories for the Apache helicopter. $M_\infty = 0.27$, $\alpha = 0.0^\circ$.

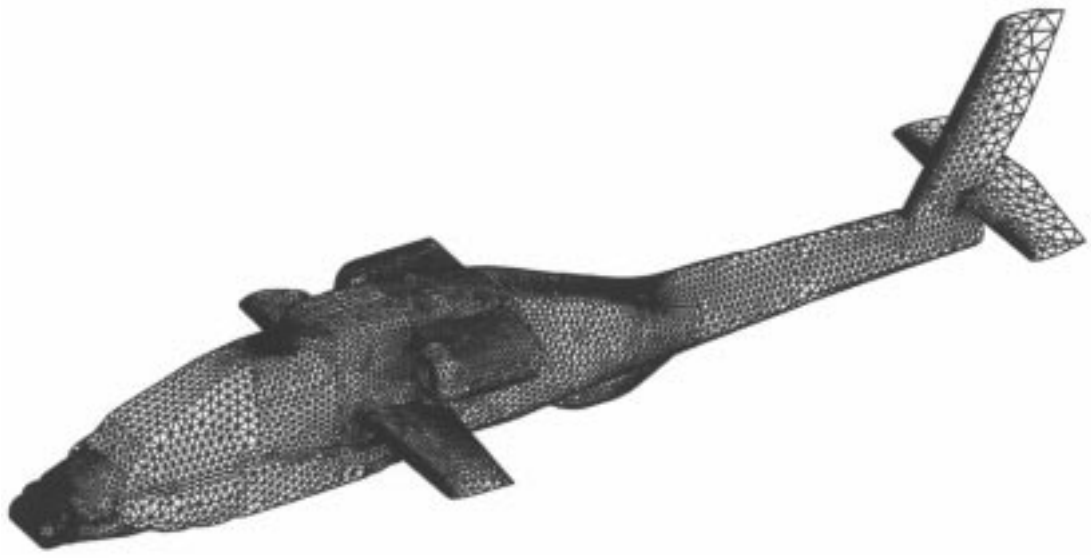

Fig. 11. Mesh for Apache helicopter configuration. (249,087 nodes)

using the Newton-Krylov scheme alone. The hybrid scheme reduces the time by over a factor of 2 for a 10 order-of-magnitude reduction of the residual. Using mesh sequencing, further improvements are gained so that almost a factor of 5 reduction in computer time is achieved compared to the Newton-Krylov scheme with no mesh sequencing. By using mesh sequencing, the computer time required compares favorably with that of the 3-level W-cycle shown in Fig. 6 and is slightly faster than using a 3-level V-cycle.

Application to Three Dimensions

Three-dimensional results are shown below for two applications. The first case is the subsonic flow over the ONERA M6 wing at a freestream Mach number of 0.699 and an angle-of-attack of 3.06° [16]. The mesh for this configuration contained 139,356 nodes. Figure 8 shows the surface grid used to model the wing geometry.

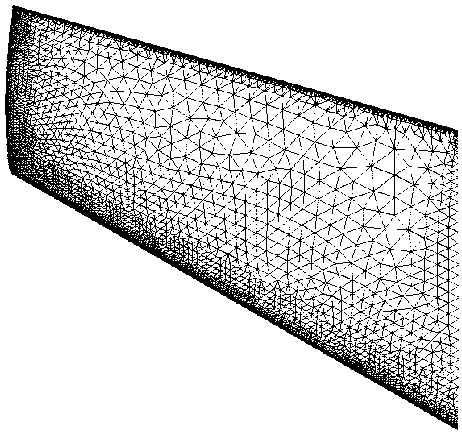

Fig. 8. Surface mesh for ONERA M6 wing.

The convergence histories for the Newton-Krylov method as well as the point Gauss-Seidel method are shown in Fig. 9. For the Newton-Krylov method, the CFL number has been increased from 20 to 10,000 as the solution converged. For the baseline scheme, the CFL is increased linearly from 20 to 100 over 100 iterations. The results in Fig. 9 show a distinct advantage of the Newton-Krylov scheme over the baseline scheme for driving the residual towards machine zero in terms of both computer time and iteration count. The method has reduced the CPU time necessary to achieve a 7 order-of-magnitude reduction by about 50 percent. Pressure distributions at various spanwise locations on the wing are shown in Fig. 10. The calculations show good agreement with experimental results.

The last case considered is the three-dimensional subsonic flow over an Apache helicopter configuration at a freestream Mach number of 0.27 and an angle-of-attack of zero degrees. The grid used for this case consists of 249,087 nodes and is shown in Fig. 11. Because of asymmetry of the aircraft, both the left and right sides of the geometry are modelled. Due to memory requirements, this case was run on the Cray-C90 located at the NASA Ames Research Center.

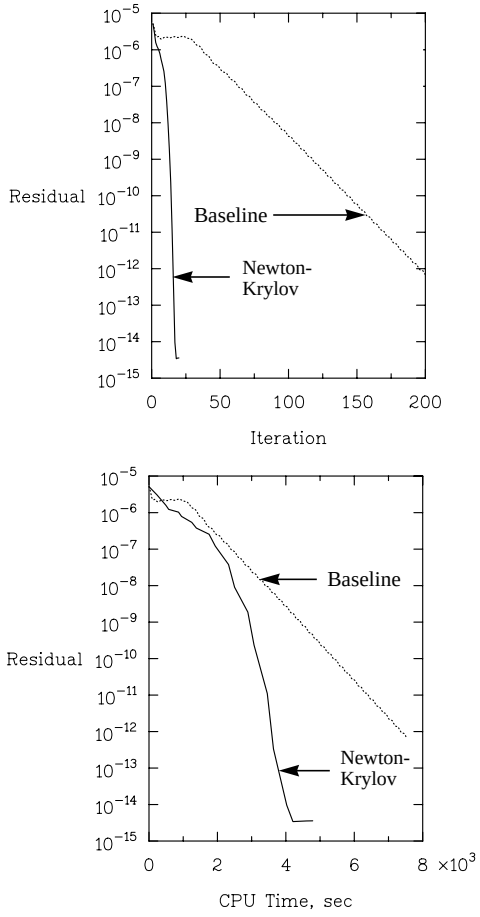

Fig. 9. Convergence histories for ONERA M6 wing.
$M_\infty = 0.699$, $\alpha = 3.06^\circ$.

For this calculation, a flux-difference splitting scheme has been used. The CFL number for the Newton-Krylov calculation is once again increased from 10 to 10,000 based on the reduction in the residual. To provide an initial solution for the second-order accurate calculations, 5 iterations using a first order accurate scheme have first been conducted. The case was also run using the baseline method, where the CFL number has been increased linearly from 10 to 200 over 100 iterations. Fig. 12 shows a plot of the residual history versus iteration count for this case. In the Newton-Krylov computations, the residual is rapidly reduced by over 2 orders of magnitude over the 5 first-order accurate iterations. The residual increases discontinuously at the point where second order accuracy is initiated. After switching to second-order accuracy, the residual decreases by 6 orders of magnitude over the next 20 iterations, at which time the computation was stopped. The baseline scheme can be seen to give very similar results in terms of the computer time.

remaining 7 orders of magnitude. This is due to the fact that the CFL number remains at a low value until the shock position stabilized and the flow field approaches the steady-state solution. For the results from the baseline scheme, the residual is reduced 10 orders of magnitude in 700 iterations which required approximately 400 seconds of computer time. However, obtaining a 2 order-of-magnitude reduction only required about 100 seconds whereas the Newton-Krylov method required about 400 seconds to achieve a comparable level of convergence, at which point the residual is then rapidly reduced.

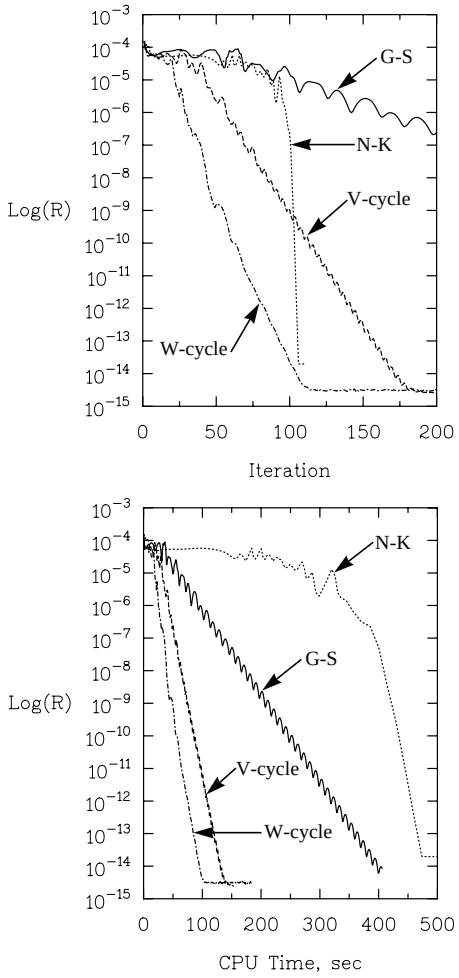

Fig. 6. Convergence histories for transonic NACA 0012.

Two methods have been investigated for improving the performance of the Newton-Krylov method caused by expending unnecessary resources in the early stages of the iterative process for transonic conditions. The first method is to use the point Gauss-Seidel scheme in the initial stages of the iteration process until the residual is reduced by a specified amount before switching to the Newton-Krylov scheme. This method is referred to as the "hybrid method" in the discussions that follow. The criterion used to establish the "switch-over" point is a two order-of-magnitude drop in the residual and is based on the results obtained above that indicate that rapid convergence will be obtained after this level is achieved. Here, the CFL number is linearly ramped from 10 to 200 during the point Gauss-Seidel calculations, and then allowed to increase during the Newton-Krylov portion of the calculation according to the residual with a maximum CFL number allowed of 10,000. The second method investigated is to obtain a partially converged solution on a coarser mesh which can be then linearly interpolated to the fine mesh in order to provide an initial solution on the fine mesh. For this calculation, 60 iterations were performed on a coarse grid of 2,298 points using the point Gauss-Seidel method. These results were then interpolated to the fine grid (8,578 points), and the iteration process was completed using the Newton-Krylov method.

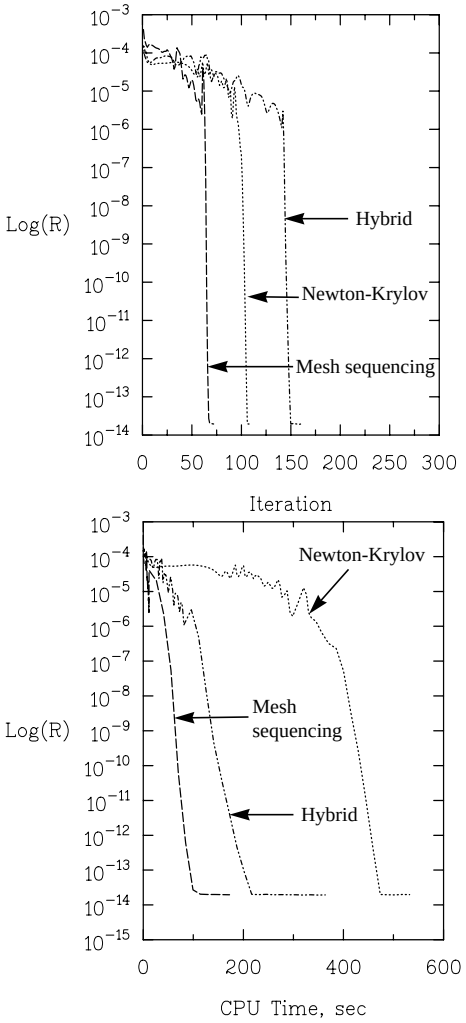

Fig. 7. Convergence histories for the Newton-Krylov method compared to a hybrid Gauss-Seidel/Newton-Krylov method and mesh sequencing for 2-D transonic airfoil.

Figure 7 shows results obtained with both the hybrid method and mesh sequencing compared to the Newton-Krylov scheme. Both the hybrid and the mesh sequencing results show a significant improvement over

this case was 10 and was increased in the same method as the solution converged. In addition, initial CFL values of 50 and 100 were also tried but diverged. As seen in the figure, the convergence of the two methods is identical except very close to machine zero where the result obtained using the exact matrix-vector product converged to a slightly lower value. The failure of the approximate method to reach exactly machine zero has been observed over a wide range of test cases and is believed to be caused by numerical inaccuracies in differencing the residual. However, it should be noted that when the approximate method does not reach machine zero, it is typically only one order of magnitude too high. Also shown in Fig. 4 is the computer time required for each of the methods. The approximate method requires slightly more computer time than using the exact linearizations because the approximate method requires an extra residual calculation. However, the use of the approximate method requires that the linearization of the residual only be stored one time for use in the preconditioner.

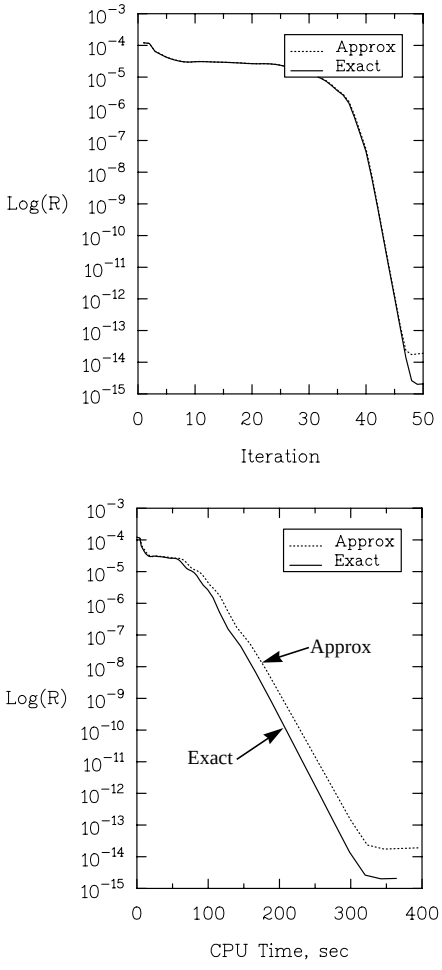

Fig. 4. Comparison of convergence rates obtained using the approximate and exact methods of forming the matrix-vector product.

As mentioned earlier, the choice of ε is crucial to the success of the Newton-Krylov method when using Eq. (15) for evaluating the matrix-vector product. The effects of choosing epsilon are illustrated in Fig. 5. Results are shown for the coarse, medium, and fine meshes. The curves labeled with symbols represent the use of a constant value of ε chosen to be the square root of machine zero ($\varepsilon = \sqrt{\varepsilon_{mach}}$). The curves without symbols denote results obtained using the present methodology of choosing ε so that the value of ε times the "typical" size of an element of the vector in the Krylov subspace is the square root of machine zero ($\varepsilon\bar{v}_j = \sqrt{\varepsilon_{mach}}$). It is clear that the latter convention consistently converges to near machine zero while choosing a constant ε fails to reach this level of convergence. Note that by choosing ε with the current method, its value increases as the mesh is refined and is not necessarily small.

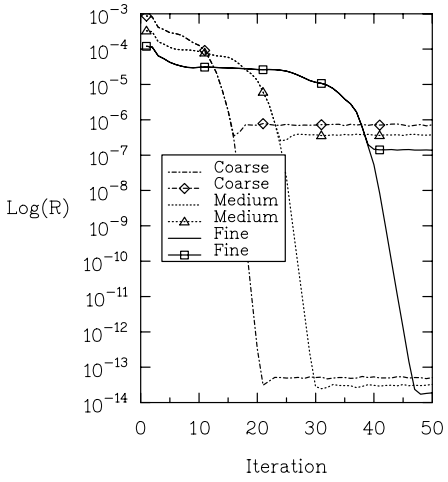

Fig. 5. Illustration of the effect of choosing ε. Curves with symbols denote a constant $\varepsilon = \sqrt{\varepsilon_{mach}}$; curves without symbols represent choosing ε so that $\varepsilon\bar{v}_j = \sqrt{\varepsilon_{mach}}$.

The second 2-D case is a transonic NACA 0012 airfoil configuration. The freestream Mach number for this case is 0.80, and the angle-of-attack is 1.25°. Results are shown in Fig. 6 for the Newton-Krylov method compared to the non-multigrid and multigrid schemes as before. For the Newton-Krylov results, the matrix-vector product is formed using Eq. (15). The CFL number has been ramped linearly from 10 to 200 over 100 iterations for the non-multigrid point Gauss-Seidel computations and ramped from 10 to 200 according to the residual for the multigrid computations. For the Newton-Krylov results, the initial CFL number of 10 is increased according to the residual and is limited to a maximum value of 10,000. The W-cycle and V-cycle multigrid computations used 5 and 7 subiterations, respectively. The Newton-Krylov method requires approximately the same number of iterations as the multigrid solution using a 3-level W-cycle. However, examination of the residual shows that approximately 100 iterations are required to obtain the first two order-of-magnitude reduction in the initial residual after which only 7 iterations are required to drop the residual the

over the entire iteration history so that the behavior of the different residuals can be observed but the actual level of convergence is identical. Figure 2 shows how each of the methods used for monitoring convergence behave over the first twenty iterations. It can be seen that the residuals based on both mass and $\Delta\rho$ have converged about one half of an order of magnitude during the first 20 iterations. This would yield a corresponding doubling of the initial CFL number. Note that the residual based on mass achieves a one-half order-of-magnitude decrease in the residual in only 5 iterations whereas the value of $\Delta\rho$ required 20 iterations to achieve this same level of reduction. However, in both cases, the residual decreases relatively slowly over the initial time steps.

The residual based on density, however, converges two orders of magnitude in the first 20 iterations. This would lead to an increase in the CFL number by a factor of over 100 times that of the initial value. While it is important to have as high a CFL number as possible, too high of a value in the initial stages can be destabilizing. On the other hand, restricting the CFL number to a low value will require many iterations to approach convergence so that the CFL number can be increased. It has been found through experimentation that using a measure of convergence based on the rate of change of density often leads to difficulty if the initial CFL number is too high because this measure decreases too rapidly in the initial stages. For this reason, the results shown in the remainder of the paper are obtained by increasing the CFL number inversely proportional to the rate of change in mass.

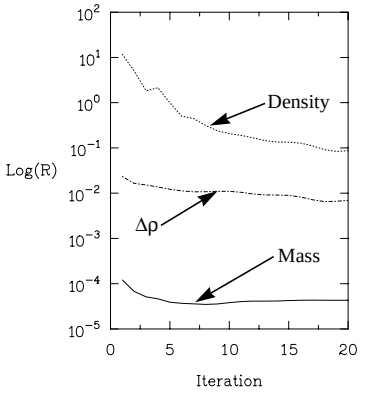

Fig. 2. Behavior of various methods of monitoring convergence.

In Fig. 3, a comparison of the convergence history in terms of both iteration count and computer time is shown for several schemes. In this figure, results obtained using the Newton-Krylov approach are compared with non-multigrid and multigrid results obtained using the baseline scheme. For the Newton-Krylov results, the matrix-vector product is computed using Eq. (15). Also, an initial CFL number of 10 is used and is increased as the solution converges with a restriction to the maximum value of 10,000. Initial values for the CFL number of 100 as well as 50 were attempted but the solution diverged after the first iteration. For the point Gauss-Seidel schemes 15 subiterations were used and the CFL number was also increased according to the residual but the maximum value was restricted to be 200. For the multigrid runs, three meshes consisting of 8,578, 2,298, and 653 nodes were used and 10 subiterations were used on each grid level.

The Newton-Krylov method shows the fastest convergence in terms of the number of iterations. In terms of CPU time, the results obtained using the 3-level W-cycle show the best convergence followed by those obtained with the V-cycle. The Newton-Krylov result requires less time to achieve a 10 order-of-magnitude reduction in residual than the non-multigridded baseline scheme. However, the Newton-Krylov scheme requires more computer time than the non-multigridded baseline scheme for the first three orders of magnitude; in many typical engineering computations, a practical solution may only require this degree of accuracy.

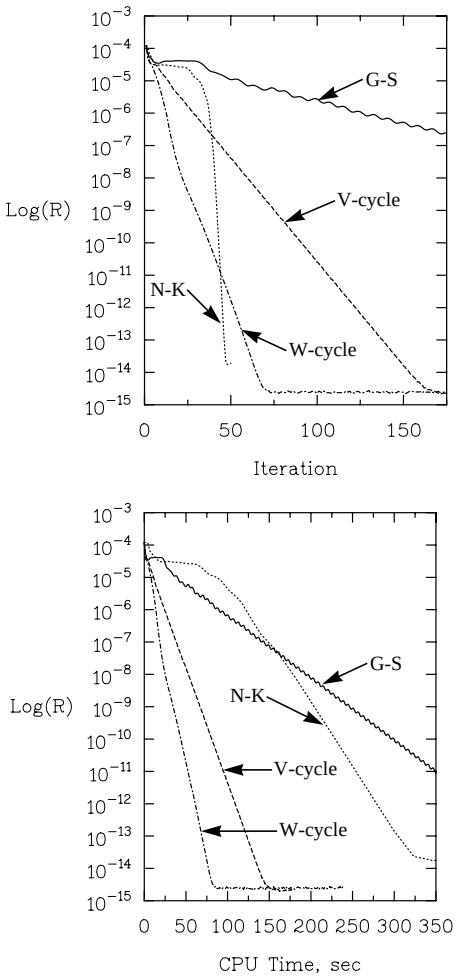

Fig. 3. Convergence histories for Subsonic NACA 0012 airfoil.

Newton-Krylov results where the matrix-vector product is obtained using the method of Ref. 4 are compared with finite-difference results in Fig. 4. As for the finite-difference results, the initial CFL number for

and is similar to that of the Gauss-Seidel scheme. Additional storage is required for saving the vectors in the Krylov subspace. The storage required for these vectors is the dimension of the Krylov subspace times the number of unknowns in the mesh. For example, with 20 search directions in a three-dimensional application, this would require 100 storage locations for each node. However, the primary storage is dictated by the preconditioner which requires that a block 5x5 (3-D) sub-matrix be stored for every non-zero element in the large matrix. For the meshes used in this work, there are roughly 14 edges that connect to each node so that the total storage (including the diagonal) for the non-zero elements in the matrix is about 375 times the number of nodes in the mesh. When evaluating the matrix-vector product by Eq. (17), the linearizations of the fluxes requires an additional amount of storage equal to that of the preconditioner, so that a total of 750 storage locations are required.

Multigrid Acceleration

For two-dimensional applications, multigrid acceleration has been incorporated into the code and is described in Ref. 5. This option can be used in conjunction with the Gauss-Seidel scheme or the Newton schemes. In addition, mesh sequencing can be used in which an initial solution is first obtained on a coarse grid and then linearly interpolated to a finer grid where it is used as an initial condition. These options provide an additional degree of freedom for reducing the total time required to converge to a steady state.

Results

Results are shown below which compare the relative efficiency of various schemes. For all these results, the scheme in which the linear system is solved using the point iterative method is referred to as the baseline scheme. For the results obtained with GMRES, 20 search directions are used and the tolerance for the linear system is such that the residual for the linear system is reduced by a maximum of three orders of magnitude. If this tolerance was not met, the algorithm was not restarted. The choice of these parameters is based on numerical experiments. It should be noted that while the tolerance could often be relaxed, which required less computer time, this approach was found to be less robust than using the tighter tolerance when using very large CFL numbers. Unless otherwise noted, all the results have been obtained on the Cray-YMP located at the NASA Langley Research Center.

Two-Dimensional

The first test case is a subsonic flow over an NACA 0012 airfoil. A near-field view of the grid used for this calculation is shown in Fig. 1 and contains 8,578 grid points with 256 of them located on the surface of the airfoil. The configuration was run at a freestream Mach number of 0.63, and an angle-of-attack of 2°.

Obtaining Newton-type convergence with the present schemes requires the use of very large CFL numbers so that the linearization term associated with the time derivative of the dependent variable vanishes, and Newton's method is obtained. However, too large an initial value of the CFL number can cause the solution to diverge. Therefore, for the applications using the Newton schemes, the CFL number is started at a reasonably small value and is allowed to increase as the solution converges so that a more globally convergent scheme is obtained. The rate at which the CFL number is increased can have a significant impact on convergence, robustness, and efficiency.

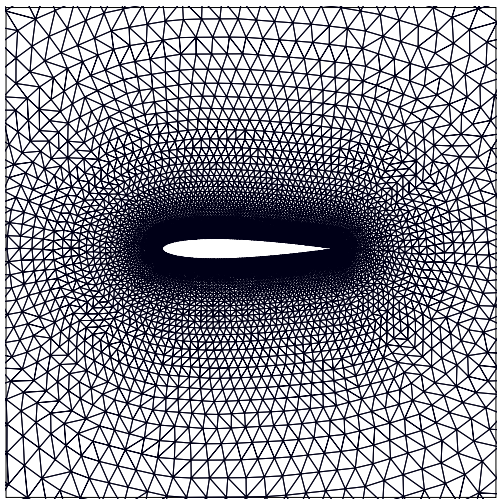

Fig. 1. Near-field view of grid for NACA 0012 airfoil.

Three methods have been considered for monitoring the convergence. The first two methods are based on the norms of

$$R_{mass} = -\oint (\vec{F}_1 \cdot \hat{n})\, d\Omega\,, \tag{18}$$

and

$$R_{dens} = -\frac{1}{V}\oint (\vec{F}_1 \cdot \hat{n})\, d\Omega\,. \tag{19}$$

These correspond to the time rate-of-change of mass and density respectively, where the integration is carried out over the surface of each control volume and includes only the contribution from the continuity equation.

The last indicator for increasing the time step as the solution converges is the change in the discrete values of the density, $\Delta\rho$. Unlike the previous methods, although a converged solution will have a $\Delta\rho$ of machine zero, its initial value depends on the initial time step. It is conceivable that for very low initial values of the CFL number, a correspondingly very low value of this parameter is obtained. In this case, its value would never decrease in relation to its initial value so that there would be no increase in the CFL number as the solution converged.

To illustrate the convergence behavior of each of these methods, the subsonic test case described earlier is used. In each case, the baseline scheme is used in which the CFL number is increased linearly with the iteration count, independent of the level of convergence. In this way, the solution variables are the same for each method

influence of nodes directly adjacent to the current node are included. With this approximation, Newton's method can only be obtained when a first-order spatial residual is used. In addition, this procedure can be slow to converge at large CFL numbers where the diagonal contribution from linearizing the time derivative of the dependent variables is small. For this reason, when this scheme is used, the maximum CFL numbers are typically restricted to numbers on the order of 100. The linearization of the residual is computed and stored along each edge and requires two matrices for each edge. In two dimensions, this requires 32 storage locations associated with each edge while in three dimensions, 50 storage locations are required.

In the remainder of this paper, this scheme is referred to as the "baseline" scheme. The performance of other schemes described below will be compared relative to this scheme.

Another method which can be used to solve the linear system of equations at each time-step is the Generalized Minimal Residual (GMRES) method.[14] The basic outline of the structure of the GMRES algorithm below is taken directly from Ref. 14.

To solve the system $Ax = b$:

1. Start: Choose an initial guess, x_0, for the solution of the linear system and compute $r_0 = b - Ax_0$ and $v_1 = \frac{r_0}{\|r_0\|}$.

2. Iterate: For $j = 1, 2, \ldots, k$ until satisfied do:

$$h_{i,j} = (Av_j, v_i), \; i = 1, 2, \ldots, j, \tag{9}$$

$$\hat{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{i,j} v_i, \tag{10}$$

$$h_{j+1,j} = \|\hat{v}_{j+1}\|, \tag{11}$$

$$\text{and } v_{j+1} = \frac{\hat{v}_{j+1}}{h_{j+1,j}}. \tag{12}$$

3. Form the approximate solution:

$$x_k = x_0 + V_k y_k, \tag{13}$$

where y_k minimizes the following functional:

$$J(y) = \left\| \beta e_1 - \bar{H}_k y \right\|. \tag{14}$$

The symbol e_1 represents the first column of the $(k+1) \times (k+1)$ identity matrix, and $\bar{H}_k$ is an upper-Hessenberg matrix with an additional row whose only nonzero element is $h_{k+1,k}$ in the $(k+1, k)$ position. The value of β is simply the norm of the initial residual $\beta \equiv \|r_0\|$.

Note from equations (9) and (10) that this procedure does not require the explicit inversion of $[A]$, but only that the product of $[A]$ with vectors v_j be computed. While the full matrix corresponding to the higher-order linearization of the residual can be formed and stored, the memory requirements are restrictive, particularly for three-dimensional flows. Therefore, instead of calculating and storing the full matrix, the matrix-vector product Av_j in equations (9) and (10) can be approximated as[6]

$$Av_j \cong \frac{R(Q + \varepsilon v_j) - R(Q)}{\varepsilon}. \tag{15}$$

where $R(Q + \varepsilon v_j)$ is the residual for the Euler equations evaluated using perturbed state quantities and ε is a scalar. As will be shown, it has been found effective to set the value of ε so that the product of ε with a "typical" element in v_j is the square root of machine zero, i.e.,

$$\varepsilon \bar{v}_j \cong \sqrt{\varepsilon_{mach}} \tag{16}$$

where ε_{mach} is the value of "machine zero" for the hardware being used (typically on the order of 10^{-14}), and $\bar{v}_j$ is the RMS value of the elements in v_j.

Replacing the matrix-vector product Av_j with a finite-difference approximation (15) of residuals discretized to second-order yields vectors v_j that correspond to those of the full second-order Jacobian but circumvents the need to compute and store the matrix. For this reason, this method will yield rapid convergence, similar to a Newton-type scheme. An example of using this approach for structured grids is given in Ref. 12 for a two-dimensional incompressible flow. For unstructured grids, Ref. 11 describes a scheme exploiting Eq. 15 for two-dimensional applications. In this reference, a diagonal preconditioner was used, as were relatively low CFL numbers, and the resulting convergence did not approach that of Newton's method.

An alternative method for computing the exact matrix-vector product for the higher-order linearization of the residual has recently been developed by Barth.[4] This scheme has the advantage that the linearization of the second-order residual is exact but the storage associated with this scheme is the same as that required when using linearizations of the lower-order residual which includes only the influence of the data at the nearest neighboring nodes. In Ref. 4, it is shown that the product Av_j can be calculated exactly as

$$Av_j = \left(\frac{\partial F}{\partial Q_L}\right) v_L + \left(\frac{\partial F}{\partial Q_R}\right) v_R. \tag{17}$$

where L and R refer to data obtained on the left and right sides of the face, respectively and is obtained using linear reconstruction. This approach to forming the matrix-vector products is also explored in this study, although heavier emphasis is given to the finite-difference technique.

Obtaining efficient results using GMRES requires the use of a preconditioner. In this study, an incomplete LU decomposition of the first-order matrix is used with no fill-in allowed [ILU(0)].[10] In addition, the bandwidth of the matrix is minimized using a reverse Cuthill-McKee algorithm.[8] The forward- and back-substitution process has been fully vectorized with a level-scheduling algorithm.[15] This is accomplished by grouping all the edges that contribute to the nodes in a current level, and coloring the edges to allow vectorization.[3,18]

The memory requirement for the Newton-Krylov approach when using the finite-difference matrix-vector products is primarily attributable to the preconditioner

MG	Multigrid
NAS	Numerical Aerodynamic Simulator
N-K	Newton-Krylov method
Q	Conserved state vector
R	Residual for a control volume
RMS	Root mean square
U	Velocity normal to the boundary of the control volume
V	Cell volume
b	A constant vector
$h_{i,j}$	Entries in H_k
k	Number of search directions
$\hat{n}$	Unit normal to a cell face
$\hat{n}_x, \hat{n}_y, \hat{n}_z$	x, y, and z components of a unit normal
p	Pressure
t	Time
u, v, w	Cartesian velocities in the x, y, and z directions
v_j	Search direction
x	Solution vector
x, y, z	Cartesian coordinates
γ	Ratio of specific heats, taken as 1.4
ε	Value used in perturbing the solution
ε_{mach}	Machine zero
η	Spanwise location
ρ	Density
Ω	Boundary of the cell

The Euler Equations

The 3-D time-dependent Euler equations for a perfect gas are given by

$$\frac{\partial Q}{\partial t} + \frac{1}{V}\oint_{\Omega} (\vec{F} \cdot \hat{n})\, d\Omega = 0 \tag{1}$$

where

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \quad \text{and} \quad \vec{F} \cdot \hat{n} = \begin{bmatrix} \rho U \\ \rho U u + \hat{n}_x p \\ \rho U v + \hat{n}_y p \\ \rho U w + \hat{n}_z p \\ (E + p)\, U \end{bmatrix}. \tag{2}$$

The component of velocity, U, in the direction of the outward-pointing unit normal to a cell face is:

$$U = \hat{n}_x u + \hat{n}_y v + \hat{n}_z w\,. \tag{3}$$

The equations are closed with the equation of state for a perfect gas:

$$p = (\gamma - 1)\left[E - \rho\frac{(u^2 + v^2 + w^2)}{2}\right]. \tag{4}$$

Solution Algorithm

The flow solver used for this study is an upwind, implicit algorithm in which the fluxes are obtained using either a flux-difference-splitting scheme,[13] or a flux-vector splitting scheme.[17] For the current algorithm, a node based scheme is used in which the variables are stored at the vertices of the grid and the equations are solved on non-overlapping control volumes surrounding each node. The solution at each time step is obtained using a backward-Euler time-differencing scheme and multigrid acceleration can be used. Further details can be found in Refs. 1 and 5.

Solution Methods for Linear System

In a second-order implicit unstructured formulation, a large system of linear equations must be solved in order to advance the solution from one time-step to the next:

$$[A]\,\{\delta Q\} = \{R\} \tag{5}$$

where

$$R = -\oint_{\Omega} (\vec{F} \cdot \hat{n})\, d\Omega \tag{6}$$

is the residual, $[A]$ is a large sparse nonsymmetric matrix

$$[A] = \frac{V}{\Delta t} I - \frac{\partial R}{\partial Q}, \tag{7}$$

and $\{\delta Q\}$ represents the change in conserved variables. The overall size of the system is the number of nodes in the mesh times the number of unknowns stored at each grid point (four in 2-D, five in 3-D).

If the complete linearization of the residual is included in $[A]$, then as $\Delta t \to \infty$, this process approaches Newton's method. However, the inversion of this linearized system by direct means requires a significant amount of memory, as well as CPU time, and is infeasible for large 3-D computations.

A simplification which significantly reduces the memory requirements for Newton's method is to replace the linearization of the residual with a linearization of a first-order accurate scheme and solve the resulting system with either an incomplete LU decomposition,[10] an iterative scheme such as a point Gauss-Seidel method, or a conjugate-gradient-type method such as GMRES[14]. However, with these simplifications, Newton-type convergence is lost.

The first method used in this study for solving the linear system of equations is a point iterative method described in Refs. 1 and 2. Here, the solution of the linear system is obtained by a relaxation scheme in which $\{\delta Q\}^n$ is obtained through a sequence of iterates $\{\delta Q\}^i$ which converge to $\{\delta Q\}^n$. In this scheme, the grid points are updated so that the odd and even numbered points are solved for alternately. In this way, a fully vectorizable Gauss-Seidel type of scheme is obtained. This can be written as

$$[D]^n \{\Delta Q\}^{i+1} = \left[\{R\}^n - [O]^n \{\Delta Q\}^{\frac{i+1}{i}}\right] \tag{8}$$

where $\{\Delta Q\}^{\frac{i+1}{i}}$ is the most recent value of ΔQ and will be at subiteration level i or $i+1$ depending on whether the current node being updated is even or odd. Here, $[D]^n$ and $[O]^n$ represent the diagonal and off-diagonal blocks of the matrix $[A]$, and $\{R\}^n$ is the vector of residuals at each grid point. When using this scheme, the linearizations of the residual are approximated by using the linearization of a first-order spatially accurate residual. In this way, only the

APPLICATION OF NEWTON-KRYLOV METHODOLOGY TO A THREE-DIMENSIONAL UNSTRUCTURED EULER CODE

Eric J. Nielsen,* W. Kyle Anderson,† Robert W. Walters‡, and David E. Keyes§

Summary

A Newton-Krylov scheme is applied to an unstructured Euler code in both two and three dimensions. A simple and computationally efficient means of differencing residuals of perturbed solutions is presented that allows consistent levels of convergence to be obtained, independent of the mesh size. Results are shown for subsonic and transonic flow over an airfoil that indicate the Newton-Krylov method can be effective in accelerating convergence over a baseline scheme provided the initial conditions are sufficiently close to the root to allow the fast convergence associated with Newton's method. Two methodologies are presented to accomplish this requirement. Comparisons are made between two methods for forming the matrix-vector product used in the GMRES algorithm. These include a matrix-free finite-difference approach as well as a formulation that allows exact calculation of the matrix-vector product. The finite-difference formulation requires slightly more computer time than the exact method, but has less stringent memory requirements. Lastly, three-dimensional results are shown for an isolated wing as well as for a complex-geometry helicopter configuration.

Introduction

Significant advancements have taken place over the past decade in the area of computational fluid dynamics, and solutions involving very complex 3-D configurations have become a reality. As geometric configurations get more detailed, the computational grids used to discretize the domain naturally become more complicated. The use of unstructured grids has some advantages over structured grids in dealing with these complicated geometries. Even in unstructured grids, the number of grid points needed to obtain accurate solutions is often very large, resulting in a large system of coupled nonlinear equations that needs to be solved. The solution process requires a large amount of computing time, and therefore it becomes advantageous to make use of the most efficient algorithms available.

One such algorithm for solving nonlinear systems is Newton's method. Unfortunately, the storage and operation cost involved in exactly solving the linear system of equations required at each time step is prohibitively expensive. An alternative approach is the Newton-Krylov method which requires neither the storage nor the inversion of the full matrix to achieve superlinear or even quadratic convergence. This is accomplished by using a conjugate-gradient type of algorithm in which the matrix-vector products are replaced with a finite-difference approximation. This algorithm has previously been successfully applied by several researchers using structured grids (e.g. Refs. 7, 12, 19), as well as to a two-dimensional unstructured Navier-Stokes code[11], and is now examined as a tool for solving three-dimensional flows.

The primary motivation for investigating the use of the Newton-Krylov method is to reduce the CPU time necessary to achieve a fully-converged solution to a given problem. By matching a second-order right-hand side with a second-order left-hand side, the method is expected to converge very rapidly, saving a significant amount of computing time.

In this study, the Newton-Krylov method is examined in both two and three dimensions to investigate its efficiency in accelerating convergence relative to currently used solution techniques. Computing time is compared with a baseline scheme, and suggestions are made to improve performance in cases that exhibit slow convergence in the initial stages of the iteration process. In addition, comparisons are made between the finite-difference technique and an exact method for evaluating matrix-vector products.

Nomenclature

A	Left-hand side matrix in implicit formulation
C_p	Pressure coefficient
CFL	Courant-Friedrichs-Lewy number
CPU	Central processing unit
E	Total energy per unit volume
$\hat{F}$	Fluxes of mass, momentum, and energy
GMRES	Generalized Minimal Residual method
G-S	Gauss-Seidel
H	Source term
H_k	Upper Hessenberg matrix
I	Identity matrix
ILU	Incomplete LU decomposition

*Graduate Student, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, Student Member, AIAA.

†Senior Research Scientist, NASA Langley Research Center, Hampton, VA 23681, Aerodynamic and Acoustic Methods Branch, Fluid Mechanics and Acoustics Division, Senior Member AIAA.

‡Professor of Aerospace & Ocean Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, Associate Fellow, AIAA.

§Associate Professor of Computer Science, Old Dominion University, Norfolk, VA 23529. Also, Senior Research Associate at ICASE, Hampton VA 23681, Member AIAA.